Unit 2 Database Practice Rockhill Music Festival

Rockhill Music Festival

Task

You have been asked to create a database for this year's Rockhill Music Festival. The festival will run over two days. The database will record information about the:

- Customers
- Tickets
- Ticket sales

There are three different types of customer. For example, a customer can be a guest of the organiser.

There are three different types of ticket:

- A Friday ticket will cost £39.00
- A Saturday ticket will cost £49.00
- A two day camping ticket will cost £88.00

Extract of Data (Figure 1)

Ticket Number	Forename	Ticket Type ID	Ticket Cost	Customer ID	Surname	Ticket Type	Telephone	Customer Type ID	Customer Type
1000	Mildred	1	£39	1	Mitchell	Friday	03415610539	3	Guest of Organiser
1001	Mildred	1	£39	1	Mitchell	Friday	03415610539	3	Guest of Organiser
1002	Amanda	1	£39	2	Ferguson	Friday	01776717391	1	Regular
1003		1	£39			Friday			
1004	Eric	2	£49	3	Ferguson	Saturday	03594633138	2	New
1005	Eric	2	£49	3	Ferguson	Saturday	03594633138	2	New
1006		2	£49			Saturday			
1007		2	£49			Saturday			
1008	Ralph	3	£88	4	Martinez	Camping	06408785372	1	Regular
1009	Ruby	3	£88	5	Butler	Camping	07975693071	3	Guest of Organiser
1010		3	£88			Camping			

Activity 1: Database relationships screenprint (45 minutes) – 8 marks

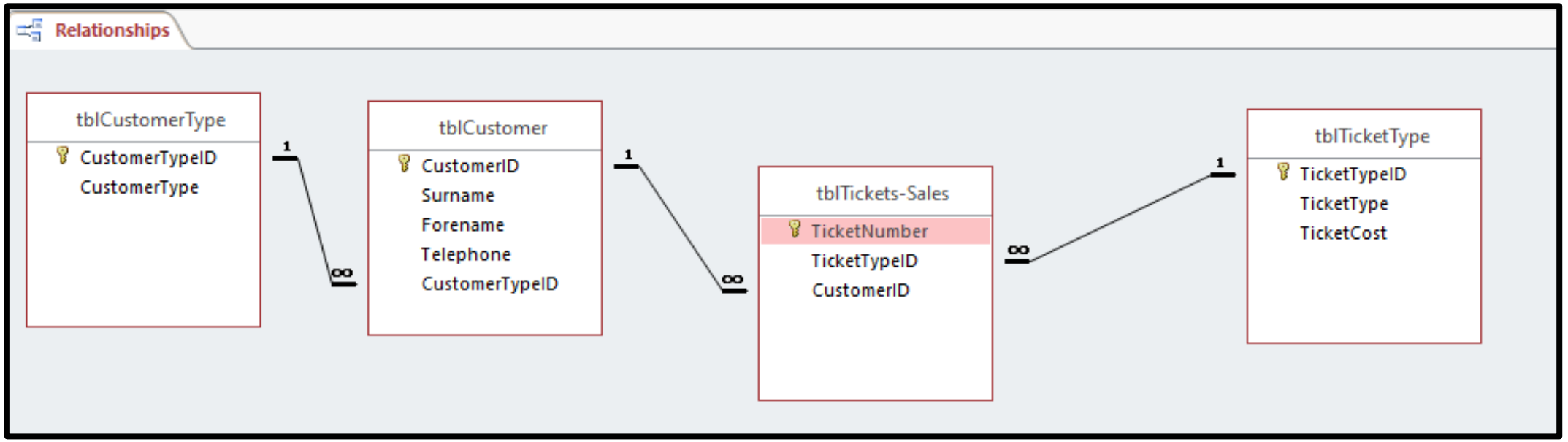
Activity 1: Database relationships screenprint (45 minutes)

Study the data extract provided in **Figure 1**. Create an efficient database structure that:
- minimises data duplication
- accepts the data provided
- uses recognised naming conventions
- ensures data integrity.

Ensure you use **all** and **only** the fields shown in **Figure 1**. Screen print your database relationships.

Save your database relationships screenprint as a PDF in your folder for submission as **activity1_[Registration number #]_[surname]_[first letter of first name]**

Activity 1: Database relationships screenprint (45 minutes) – 8 marks



Activity 1: Database relationships screenprint (45 minutes) – 8 marks - markscheme

Assessment focus	Band 0	Band 1	Band 2	Band 3	Band 4	Max. mark
Activity 1: ERD screenprint	**0**	**1-2**	**3-4**	**5-6**	**7-8**	**8**
	No rewardable material	ERD shows an attempt at normalisation with significant data redundancy. ERD has some correct relationships shown.	ERD shows that most data is correctly normalised with some data redundancy. ERD has some correct relationships and some correct relationship types.	ERD shows that most data is correctly normalised with minimal data redundancy. ERD has mostly correct relationships and mostly correct relationship types shown.	The ERD shows that the data is correctly normalised with no data redundancy. ERD has correct relationships and relationship types shown throughout.	

Activity 2: Table structures and validation (45 minutes) – 8 marks

Create an efficient table structure based on Activity 1 and the data shown in **Figure 1**. The table structures must use suitable validation to meet these requirements:

- a record will not save without the customer's surname being present
- a record will not save if the customer telephone number is not in the correct format
- a record will not save if the customer is assigned an invalid customer type
- a record will not save if the cost of a ticket is not one of the three permitted values
- a record will not save if a ticket sale does not have a valid customer
- a record will not save if a ticket sale does not have a valid ticket type.

Input the data given in **Figure 1** into your relational database.

Validation including one suitable example for each of these:

- presence check
- length check
- value lookup
- table lookup
- format check.

Activity 2: Table structures and validation (45 minutes) – 8 marks

tblCustomer

Field Name	Data Type
CustomerID	Number
Surname	Short Text
Forename	Short Text
Telephone	Short Text
CustomerTypeID	Number

tblCustomerType

Field Name	Data Type
CustomerTypeID	AutoNumber
CustomerType	Short Text

tblTickets-Sales

Field Name	Data Type
TicketNumber	Number
TicketTypeID	Number
CustomerID	Number

tblTicketType

Field Name	Data Type
TicketTypeID	AutoNumber
TicketType	Short Text
TicketCost	Currency

Activity 2: Table structures and validation – 8 marks - answers

- a record will not save without the customer's surname being present **(Presence Check)**

tblCustomer

Field Name	Data Type
CustomerID	Number
Surname	Short Text

General	Lookup
Field Size	50
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	Is Not Null
Validation Text	Please Enter a Customer Surname
Required	Yes
Allow Zero Length	Yes

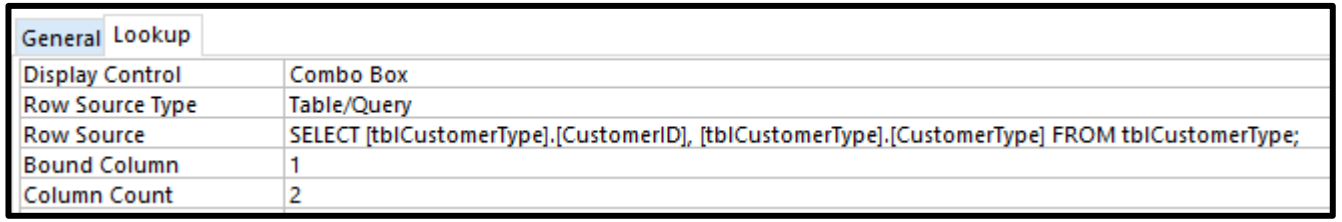
- a record will not save if the customer telephone number is not in the correct format **(Format Check)**

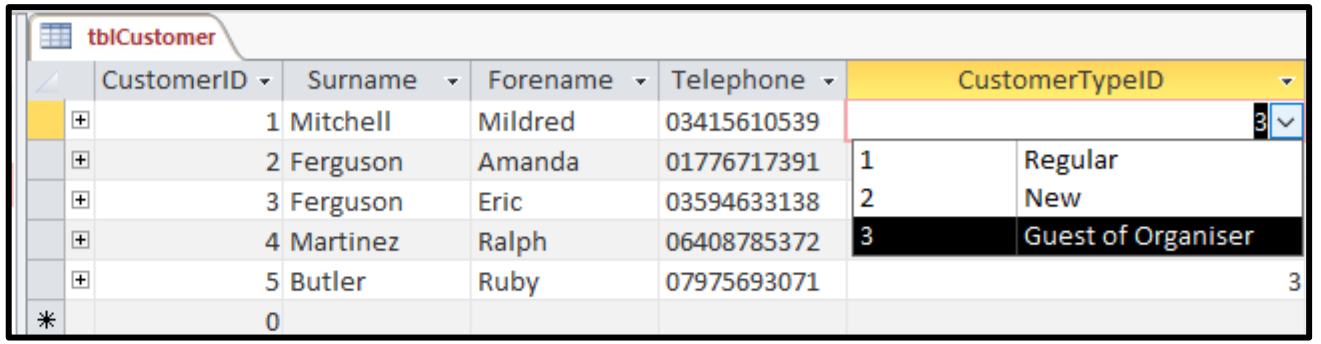
Forename	Short Text
Telephone	Short Text
CustomerTypeID	Number

General	Lookup
Field Size	11
Format	
Input Mask	00000000000
Caption	
Default Value	
Validation Rule	Is Not Null And Like '###########'
Validation Text	An 11 digit telephone number is required
Required	No
Allow Zero Length	Yes

Activity 2: Table structures and validation – 8 marks - answers

- a record will not save if the customer is assigned an invalid customer type **(Table Lookup on foreign key)**





Activity 2: Table structures and validation – 8 marks - answers

- a record will not save if the cost of a ticket is not one of the three permitted values **(Format Check)**

tblTicketType

Field Name	Data Type
TicketTypeID	AutoNumber
TicketType	Short Text
TicketCost	Currency

General | Lookup

Format	Currency
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	39 Or 49 Or 88
Validation Text	Ticket cost can only be £39, £49 or £88

- a record will not save if a ticket sale does not have a valid customer **(Table Lookup on foreign key)**

tblTickets-Sales

Field Name	Data Type
TicketNumber	Number
TicketTypeID	Number
CustomerID	Number

tblTicketType | tblTickets-Sales

TicketNumb	TicketTypeII	CustomerID	Click to Add
1000	1	1	
1001	1		
1002	1		
1003	1		
1004	2		
1005	2		
1006	2		

5	Butler	Ruby
3	Ferguson	Eric
2	Ferguson	Amanda
4	Martinez	Ralph
1	Mitchell	Mildred

General | Lookup

Display Control	Combo Box
Row Source Type	Table/Query
Row Source	SELECT [tblCustomer].[CustomerID], [tblCustomer].[Surname], tblCustomer.[Forename] FROM tblCustomer ORDER BY [Surname];
Bound Column	1
Column Count	3

Activity 2: Table structures and validation – 8 marks - answers

- a record will not save if a ticket sale does not have a valid ticket type **(Value Lookup)**

tblTicketType

Field Name	Data Type
TicketTypeID	AutoNumber
TicketType	Short Text
TicketCost	Currency

General | Lookup

Display Control	Combo Box
Row Source Type	Value List
Row Source	"Friday";"Saturday";"Camping"
Bound Column	1
Column Count	1
Column Heads	No
Column Widths	2.54cm
List Rows	16
List Width	2.54cm
Limit To List	Yes

tblTicketType

TicketTypeII	TicketType	TicketCost
1	Friday	£39.00
2		£49.00
3		£88.00
(New)		

Friday
Saturday
Camping

Activity 2: Table structures and validation (45 minutes) – 8 marks - markscheme

Activity 2: Table structure and validation	**0**	**1-2**	**3-4**	**5-6**	**7-8**	**8**
	No rewardable material	Uses some meaningful field and table names with some inconsistencies. The table structure identifies some primary and foreign key fields. The table structure has limited use of correct data types. Limited use of validation which may be inaccurate.	Uses meaningful field and table names with minor inconsistencies. The table structure identifies most primary and foreign key fields. The table structure has correct data types for most fields. Accurate validation rules for some of the fields that require validation.	Uses a recognised naming convention with minor inconsistencies for fields and tables. The table structure identifies all primary and most foreign key fields. The table structure has correct data types for most fields including matching primary and foreign key fields. Accurate validation rules for most of the fields that require validation.	Uses a recognised naming convention consistently for fields and tables. The table structure identifies all primary and foreign key fields. The table structure has correct data types for all fields. Accurate validation rules for all fields that require validation.	

Activity 3: Queries and Report (40 minutes) – 12 marks

Queries

(a) Create a query to display an alphabetically sorted list of regular and new customers. It must show the customer name and telephone number only.

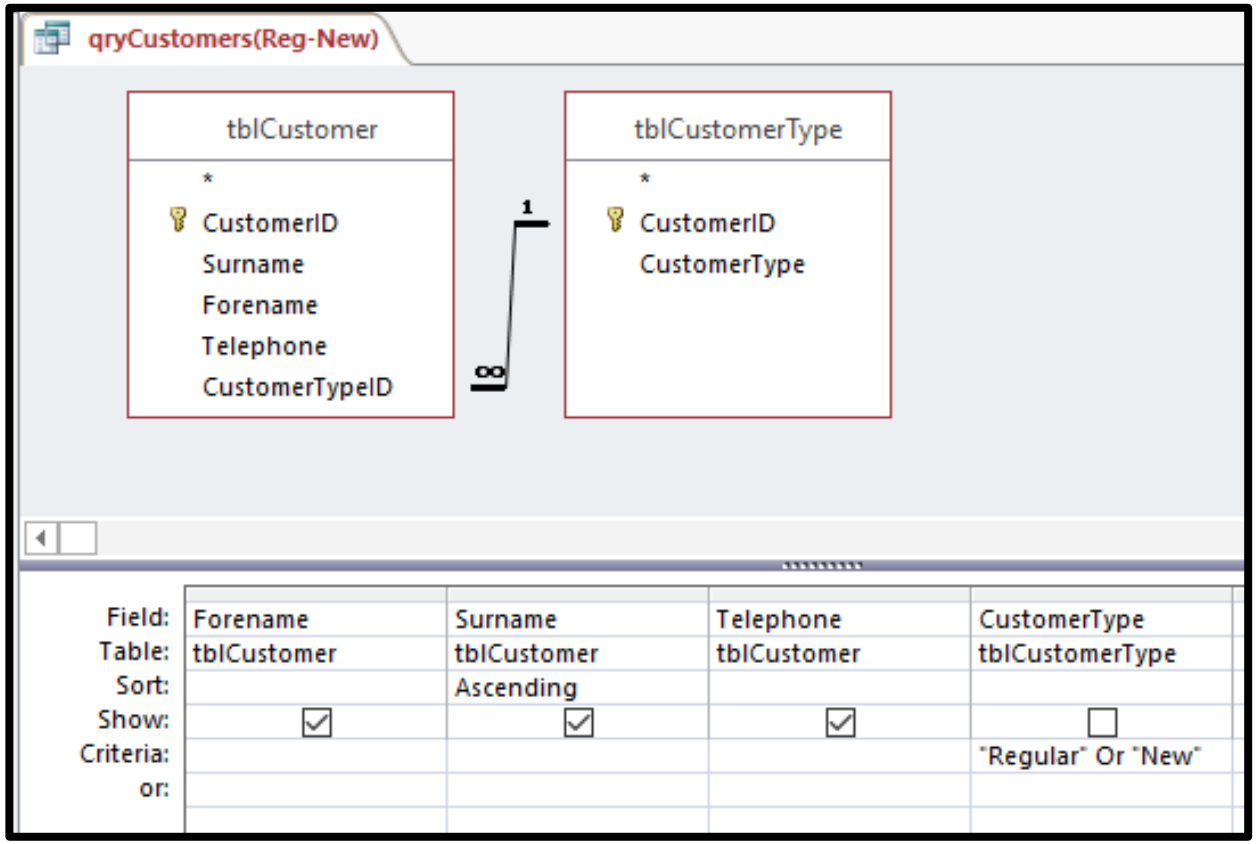
(b) Create a query that would allow a user to enter a parameter value for the ticket type when run. Calculate and display the:

- number of tickets unsold
- potential income from unsold tickets.

Activity 3: Queries and Report (40 minutes) – 12 marks

Queries

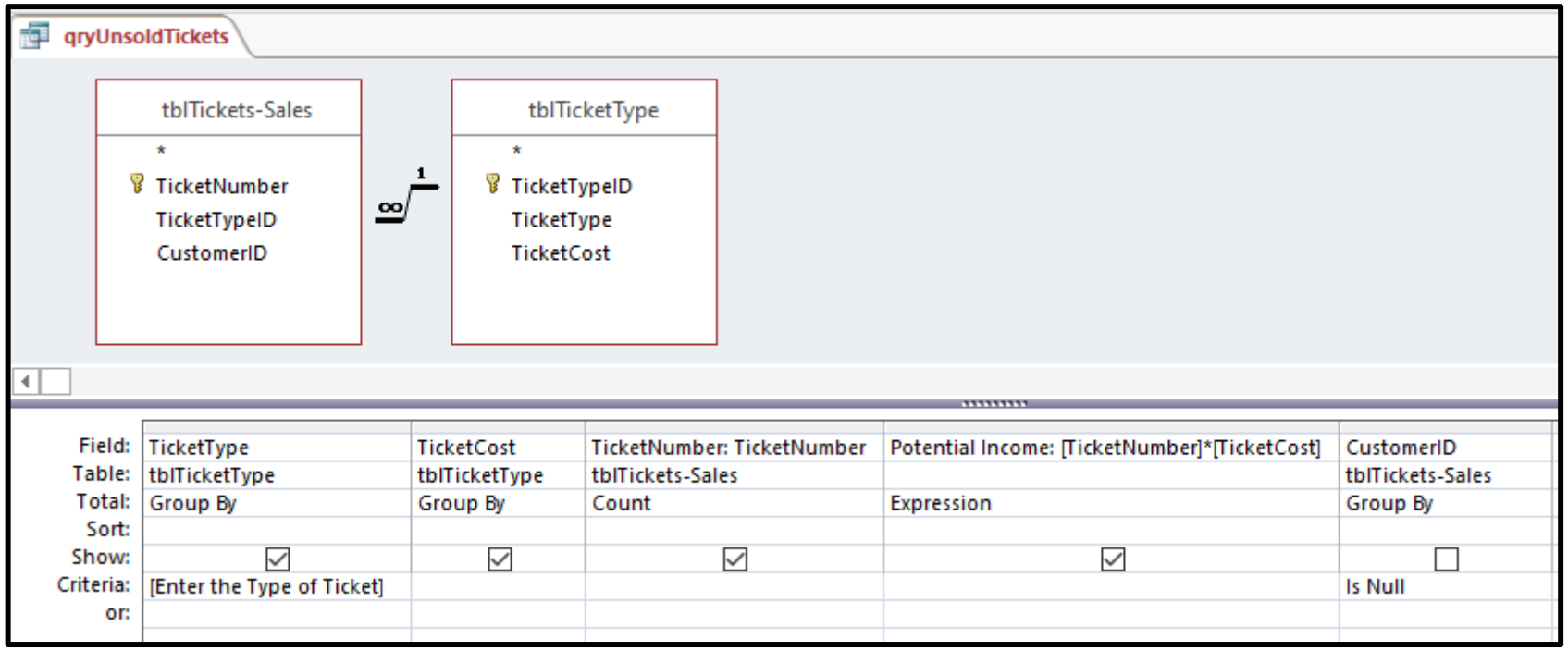
(a) Create a query to display an alphabetically sorted list of regular and new customers. It must show the customer name and telephone number only.



Activity 3: Queries and Report (40 minutes) – 12 marks

(b) Create a query that would allow a user to enter a parameter value for the ticket type when run. Calculate and display the:

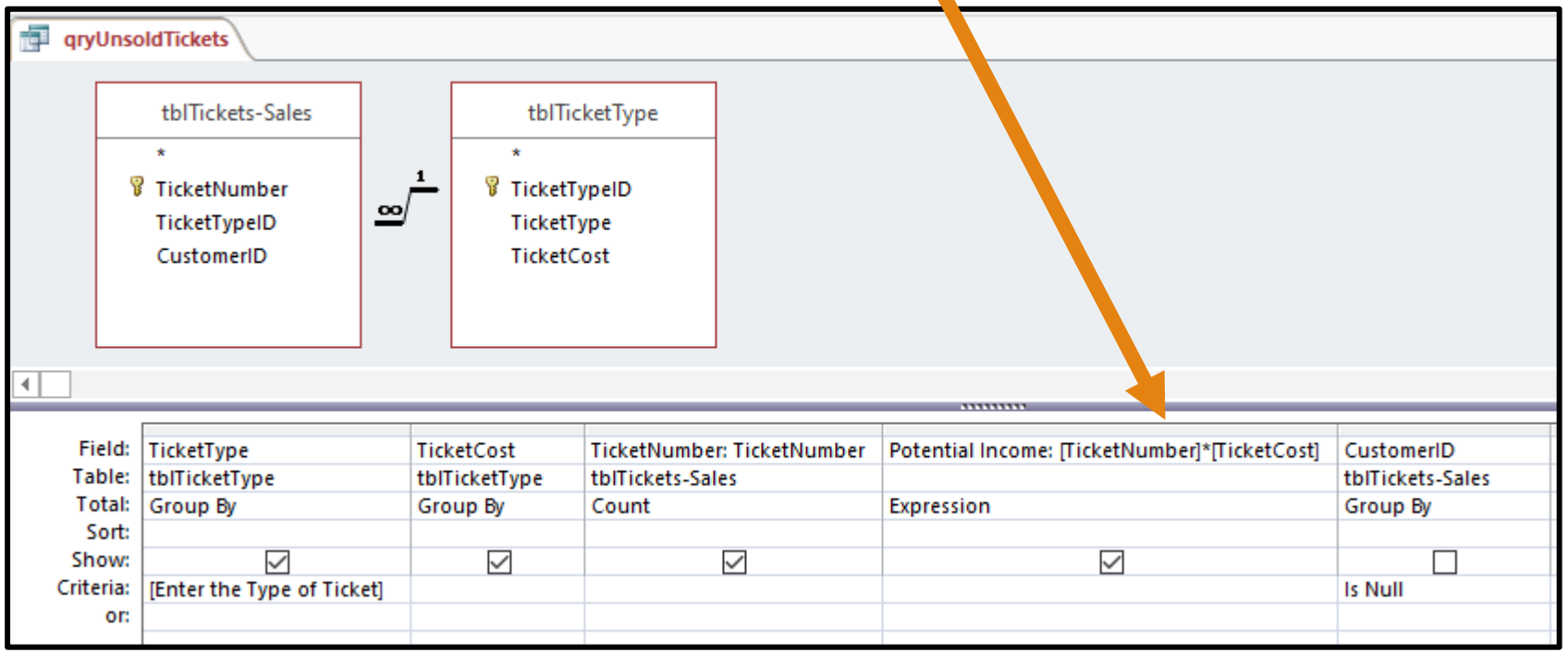
- number of tickets unsold
- potential income from unsold tickets.

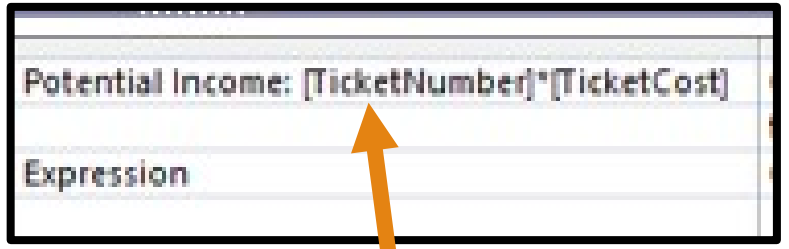


Field:	TicketType	TicketCost	TicketNumber: TicketNumber	Potential Income: [TicketNumber]*[TicketCost]	CustomerID
Table:	tblTicketType	tblTicketType	tblTickets-Sales		tblTickets-Sales
Total:	Group By	Group By	Count	Expression	Group By
Sort:					
Show:	☑	☑	☑	☑	☐
Criteria:	[Enter the Type of Ticket]				Is Null
or:					

Using the ‘Expression Builder’ to create calculations

It is not necessary to memorise the formulas required for calculations or ‘Expressions’ – you can if you wish but errors are often made if you mis-type something. Practice using the ‘Build’ Function to avoid this type of error.

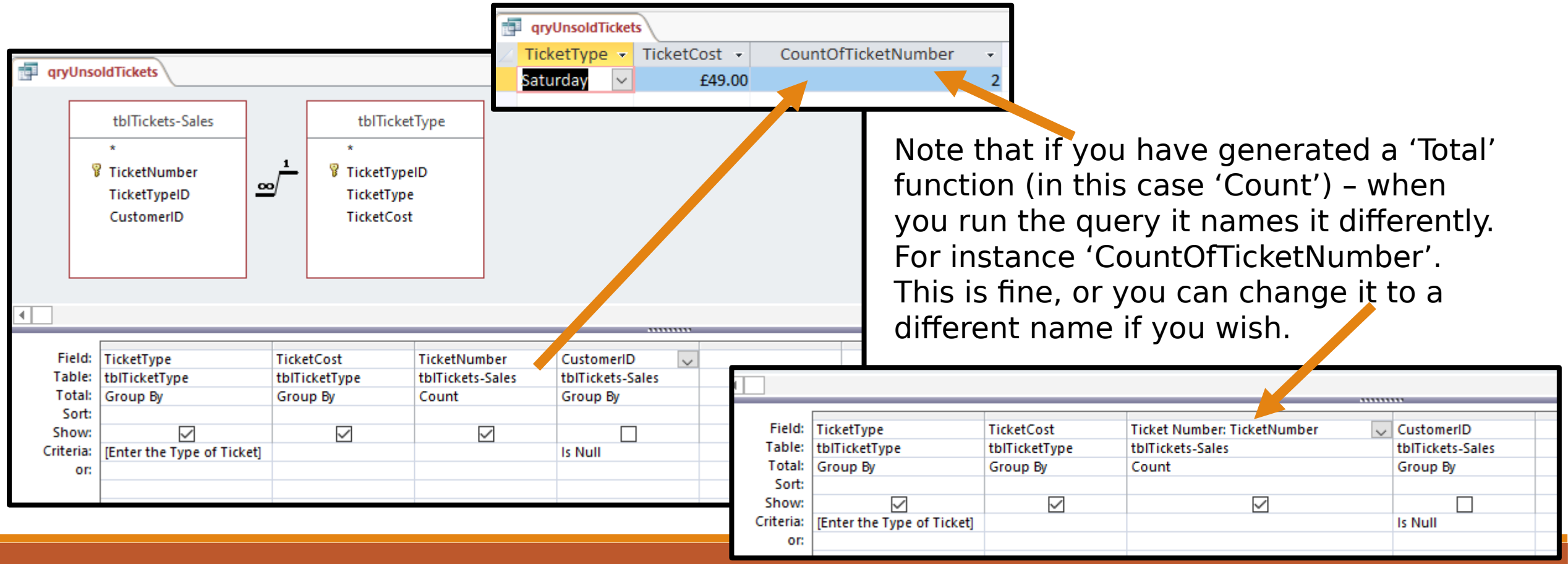




The following slides show you how to use the ‘Build’ function to generate this expression. Note that it refers to two other named fields in this query.

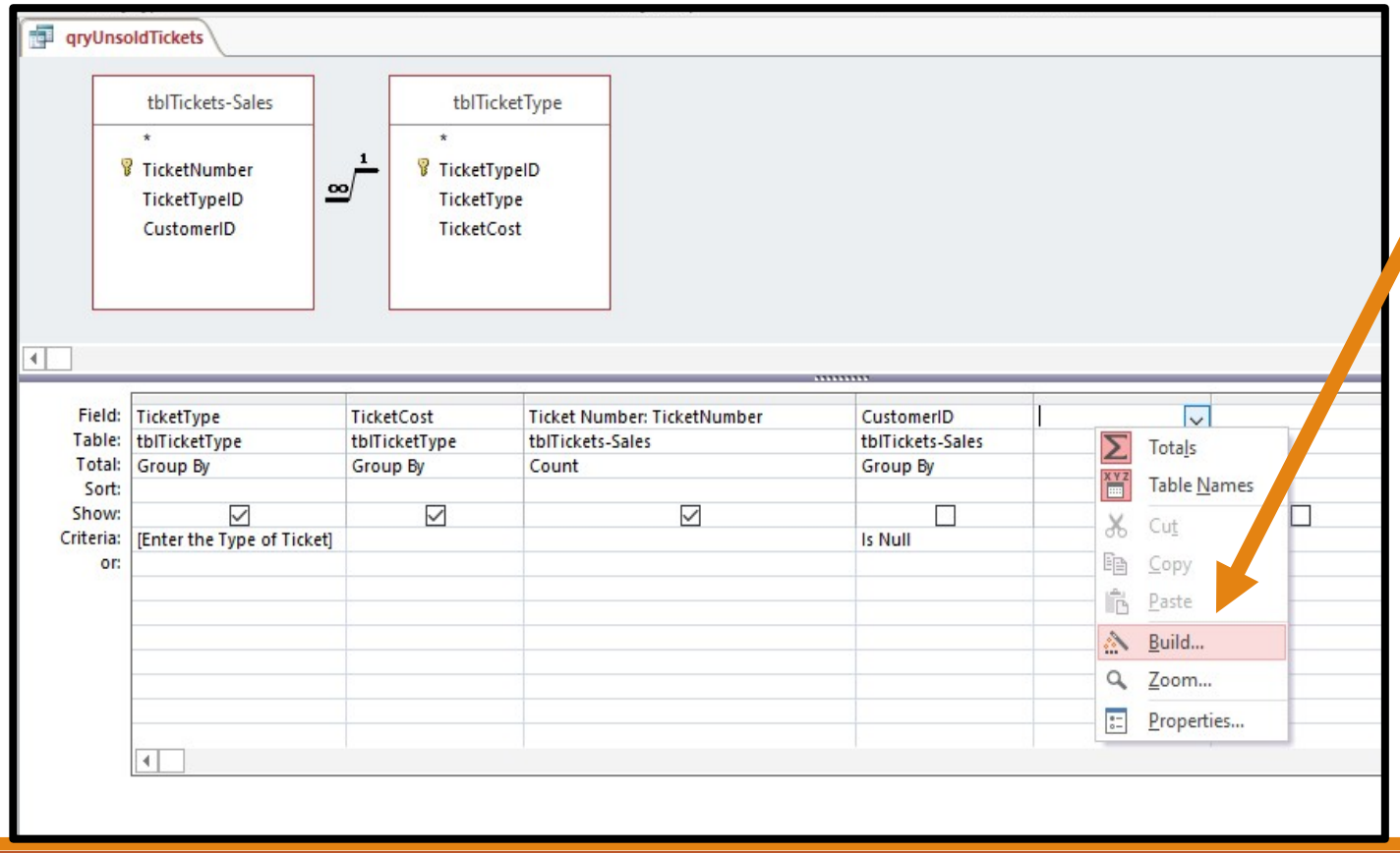
Using the ‘Expression Builder’ to create calculations

Step 1: First of all, build all of the query **without** the calculated fields that refer to each other. It is OK to include simple calculations in the ‘Total’ row such as sum and count. You must **save** before you go on to build any further calculations.

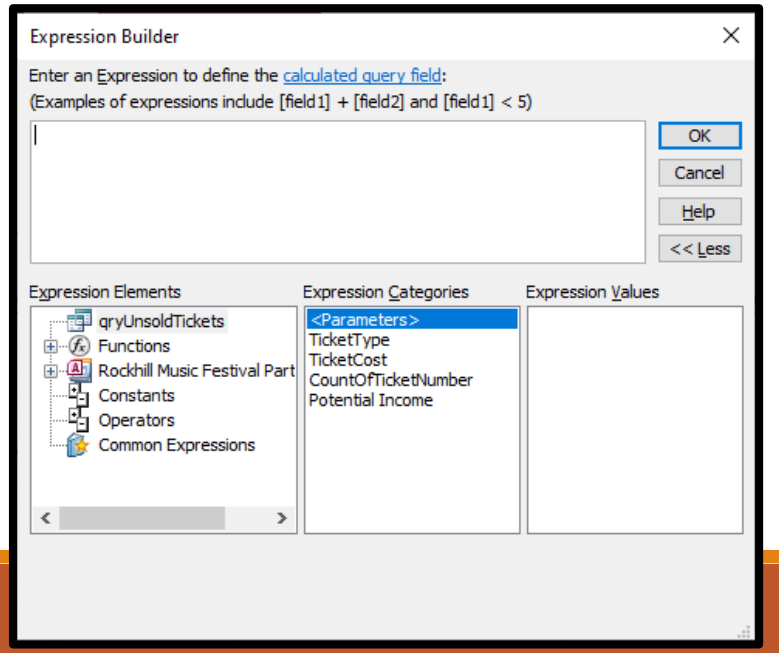

qryUnsoldTickets
TicketType
TicketCost
CountOfTicketNumber
Saturday
£49.00
2
qryUnsoldTickets
tblTickets-Sales
*
TicketNumber
TicketTypeID
CustomerID
tblTicketType
*
TicketTypeID
TicketType
TicketCost
Field: TicketType | TicketCost | TicketNumber | CustomerID
Table: tblTicketType | tblTicketType | tblTickets-Sales | tblTickets-Sales
Total: Group By | Group By | Count | Group By
Sort:
Show:
Criteria: [Enter the Type of Ticket] | | | Is Null
or:
Note that if you have generated a ‘Total’ function (in this case ‘Count’) – when you run the query it names it differently. For instance ‘CountOfTicketNumber’. This is fine, or you can change it to a different name if you wish.
Field: TicketType | TicketCost | Ticket Number: TicketNumber | CustomerID
Table: tblTicketType | tblTicketType | tblTickets-Sales | tblTickets-Sales
Total: Group By | Group By | Count | Group By
Sort:
Show:
Criteria: [Enter the Type of Ticket] | | | Is Null
or:

Using the ‘Expression Builder’ to create calculations

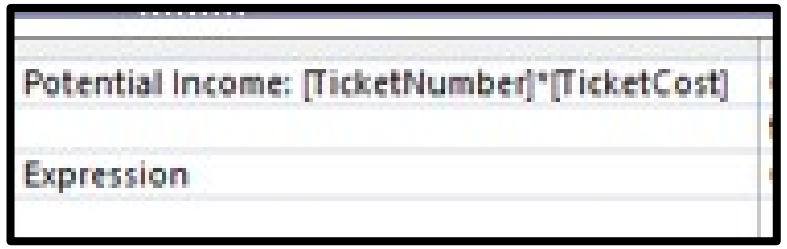
We now want to generate this ‘Expression’ or calculation. Note that in this instance we are multiplying one field by another.



Right click in the ‘Field’ cell that you want the expression to go in. The Expression Builder window will open.



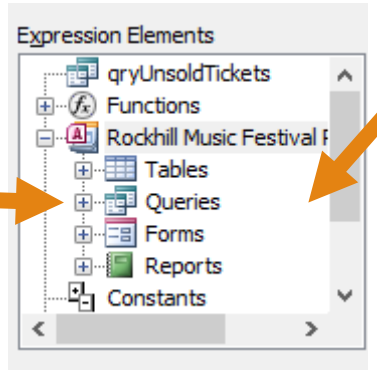
Using the ‘Expression Builder’ to create calculations

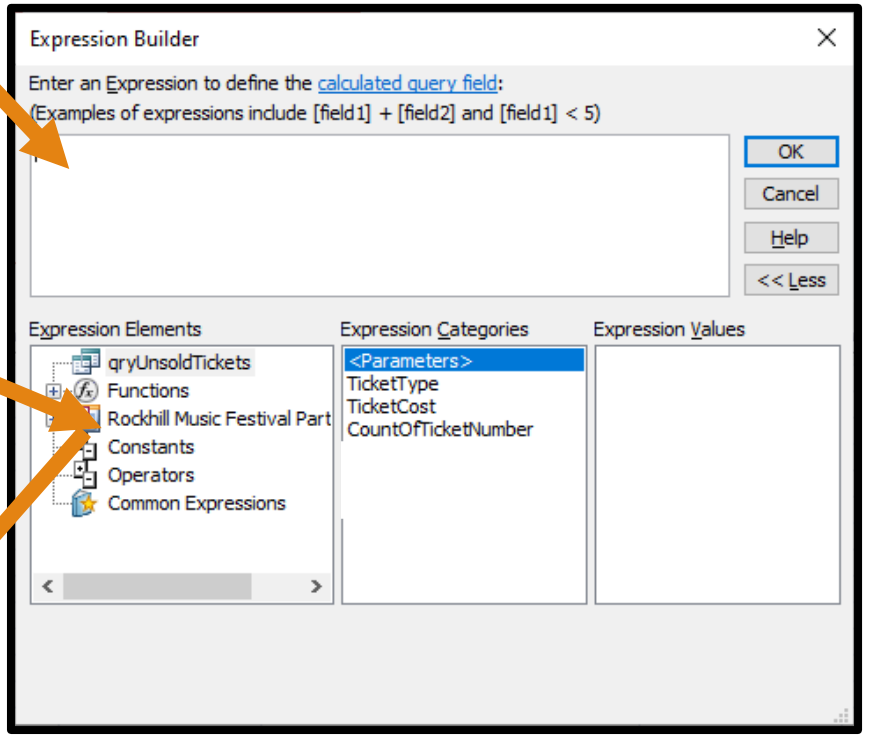


You will now ‘Build’ the expression in the white space using the Expression Elements underneath

You will see here all the components that you have already built in your database – including this current query if you have saved it. That is why saving the query where you want to build the expression is so important.

Double click the database name (Rockhill Music Festival) or click the plus sign to expand and see the elements of it.

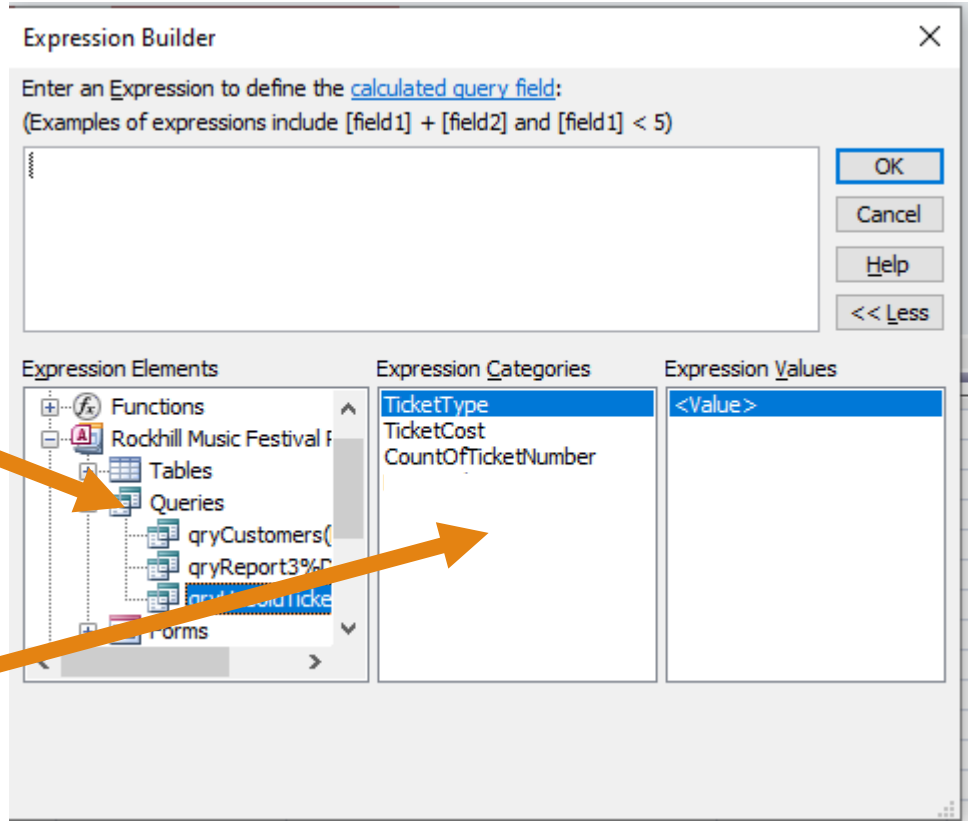




Using the ‘Expression Builder’ to create calculations

We want to build an expression from our current query, so expand the Queries element to see all of the queries available to us.

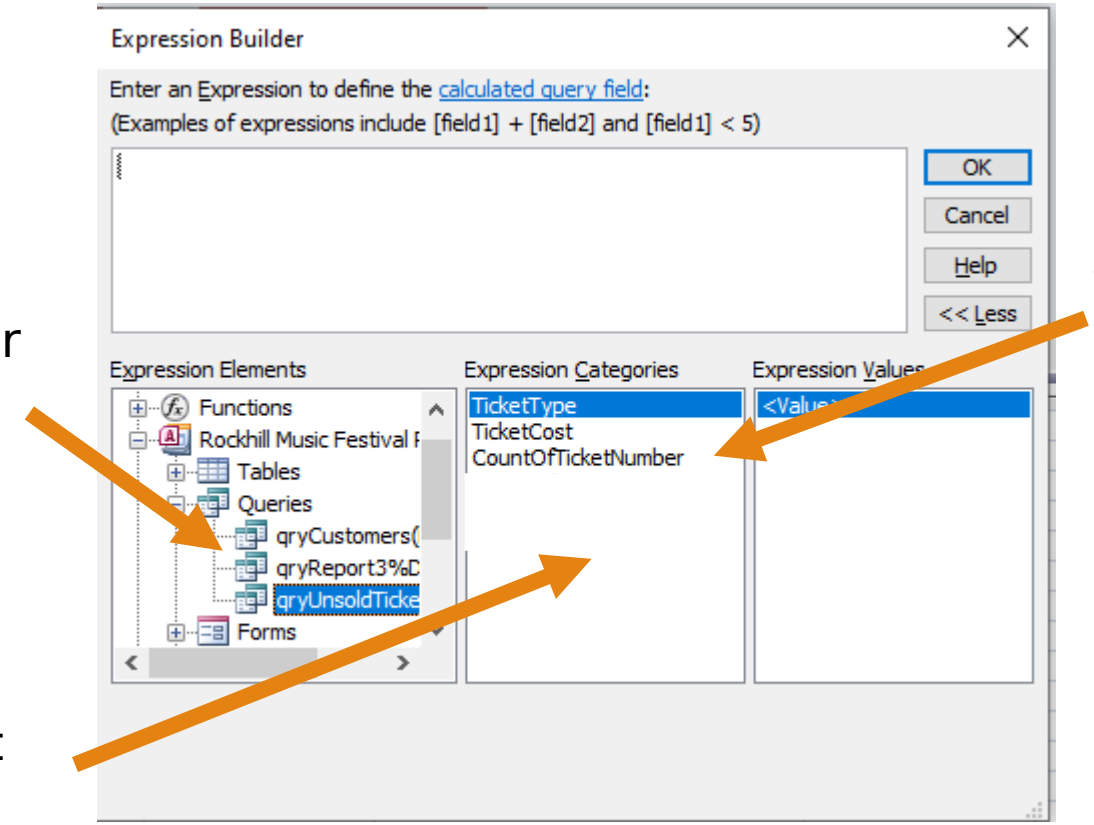
Choose the current query ‘qryUnsoldTickets’ – once you click on it you will see the fields in that query appear in the ‘Expression Categories column’.



You can simply double click on the ones that you want to appear in your formula, separated by the operator (e.g. +, -, *)

Using the ‘Expression Builder’ to create calculations

We want to build an expression from our current query, so expand the Queries element to see all of the queries available to us.



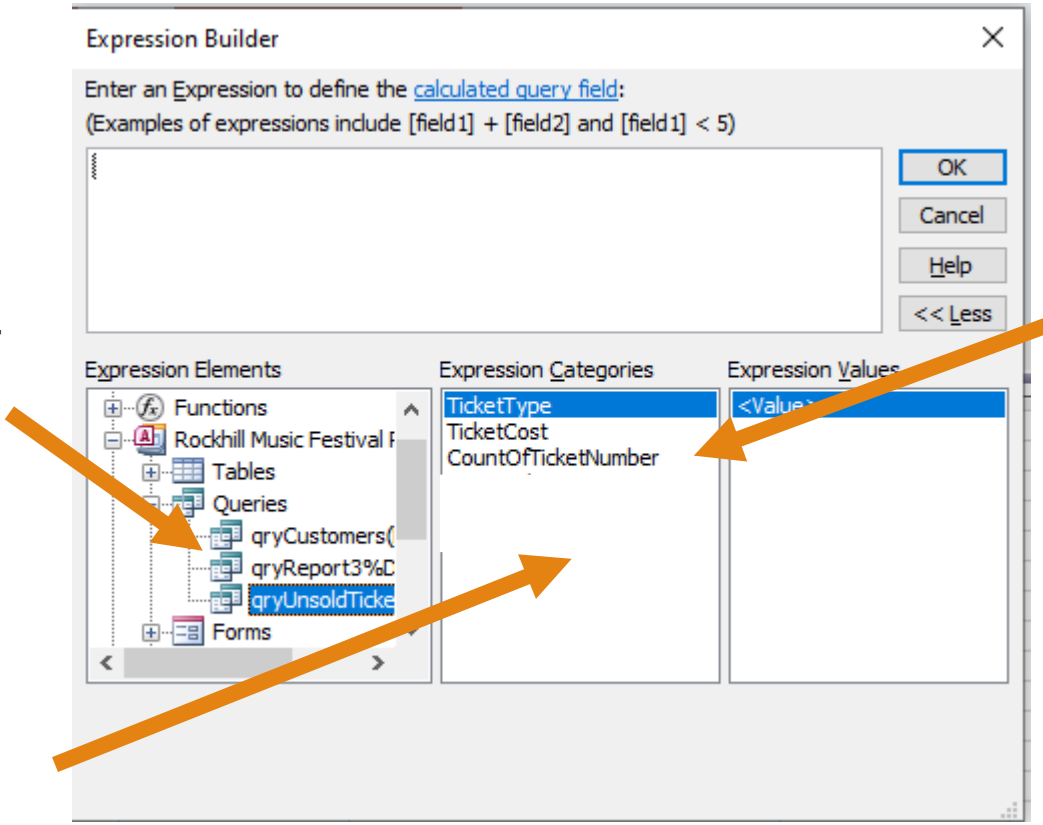
Note that CountOfTicketNumber is appearing here. This is because I failed to resave after renaming it. See next slide where this changes after an update of the save.

Choose the current query ‘qryUnsoldTickets’ – once you click on it you will see the fields in that query appear in the ‘Expression Categories column’.

You can simply double click on the ones that you want to appear in your formula, separated by the operator (e.g. +, -, *)

Using the ‘Expression Builder’ to create calculations

We want to build an expression from our current query, so expand the Queries element to see all of the queries available to us.



Note that CountOfTicketNumber is appearing here. This is because I failed to resave after renaming it. See next slide where this changes after an update of the save.

Choose the current query ‘qryUnsoldTickets’ – once you click on it you will see the fields in that query appear in the ‘Expression Categories column’.

You can simply double click on the ones that you want to appear in your formula, separated by the operator (e.g. +, -, *)

Using the ‘Expression Builder’ to create calculations

Double click on ‘Ticket Number

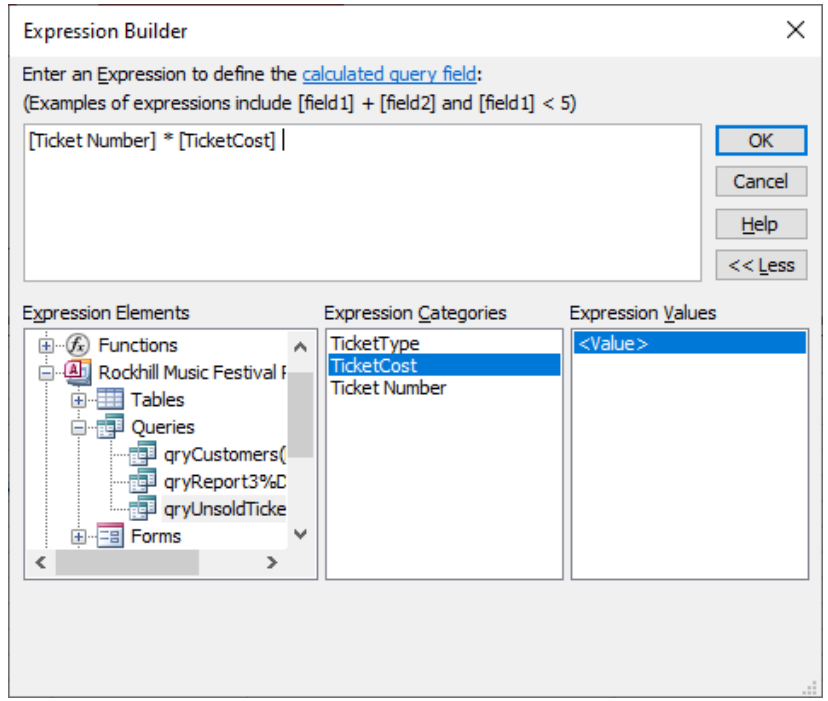
Type a * (for multiply)

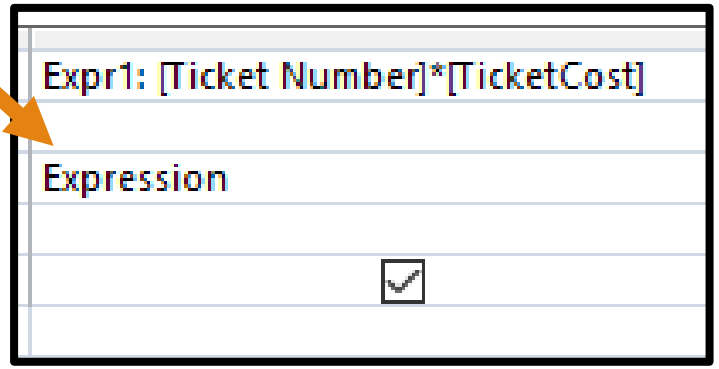
Double click on TicketCost

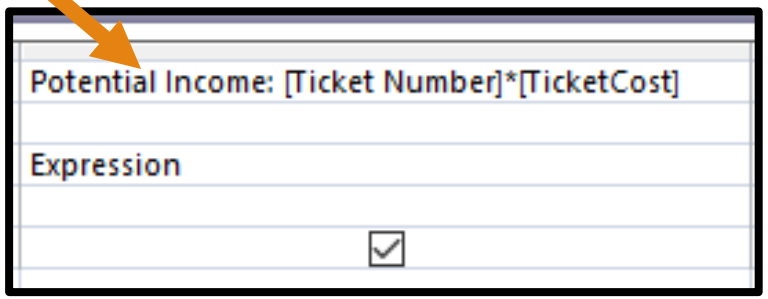
Square brackets will automatically be entered

Click ‘OK’

Choose ‘Expression in the Total row and change the name







Activity 3: Queries and Report (40 minutes) – 12 marks

(c) Create a report that shows the effect of having a 3% discount on the ticket price for
tickets that have been sold.

Calculate:
- the original income from ticket sales
- the potential discount
- the discounted ticket sales.

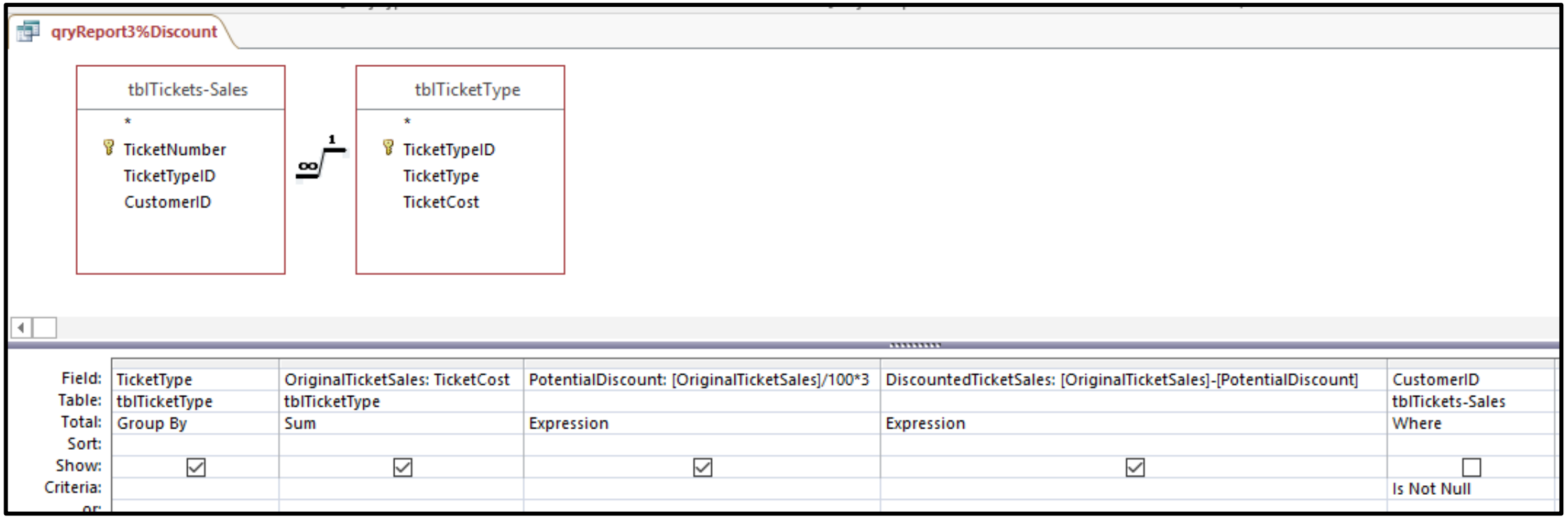
Display:
- a suitable report title
- the ticket types
- the original ticket sales
- the potential discount
- the discounted ticket sales.

The report must fit on one page.

Calculate:
- the original income from ticket sales
- the potential discount
- the discounted ticket sales.

Display:
- a suitable report title
- the ticket types
- the original ticket sales
- the potential discount
- the discounted ticket sales.

Query:



Calculate:

- the original income from ticket sales
- the potential discount
- the discounted ticket sales.

Display:

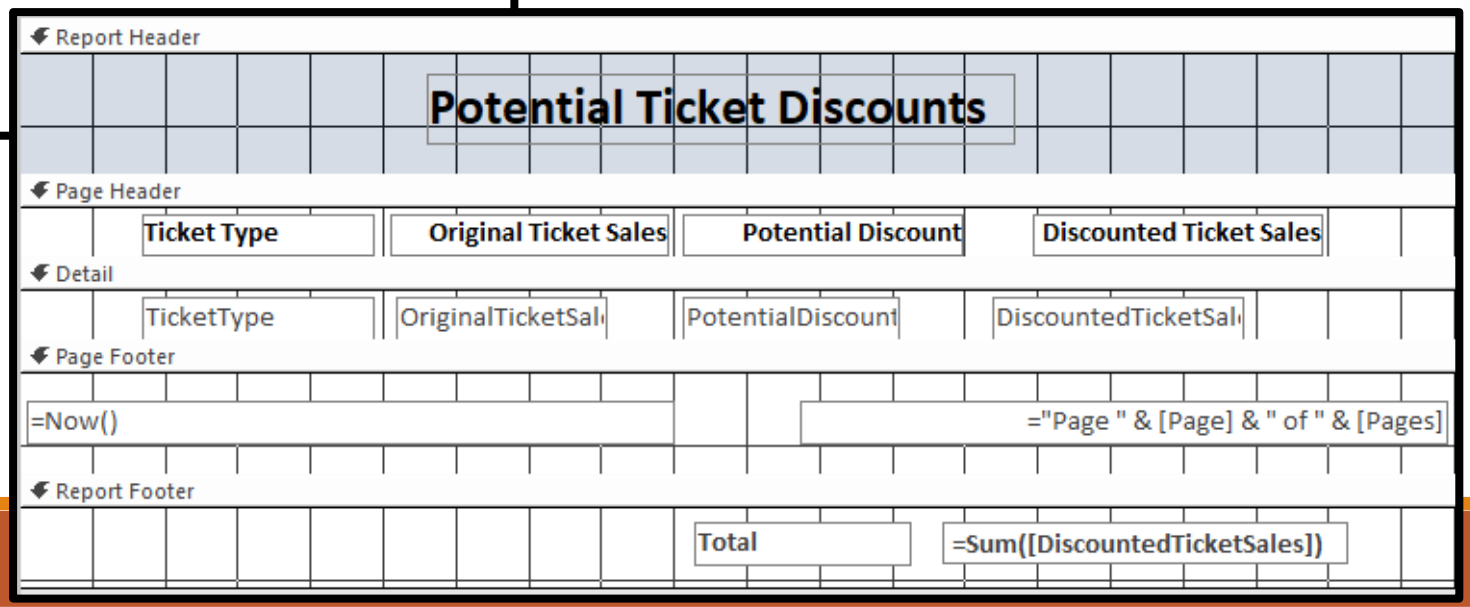
- a suitable report title
- the ticket types
- the original ticket sales
- the potential discount
- the discounted ticket sales.

Report:

Potential Ticket Discounts

Ticket Type	Original Ticket Sales	Potential Discount	Discounted Ticket Sales
Camping	£176.00	£5.28	£170.72
Friday	£117.00	£3.51	£113.49
Saturday	£98.00	£2.94	£95.06
		Total	£379.27

08 November 2022



Activity 3: Queries and Report (40 minutes) – 12 marks - markscheme

Assessment focus	Band 0	Band 1	Band 2	Band 3	Band 4	Max. mark
Activity 3: Queries and Report	0	1-3	4-6	7-9	10-12	12
	No rewardable material	Queries and report include limited relevant fields. Queries and report include details of some criteria and calculations required, which may include inaccuracies. Presentation of data in queries and report will not aid readability and understanding of data.	Queries and report includes some relevant fields. Queries and report include accurate details of some criteria and calculations required. Presentation of data in queries and report will, in places, aid readability of and understanding of data.	Queries and report includes mostly relevant fields. Queries and report includes accurate details of most criteria and calculations required. Presentation of data in queries and report will mostly aid readability and understanding of data.	Queries and report includes all relevant fields only. Queries and report include accurate details of all criteria and calculations required. Presentation of data in queries and report will aid readability and understanding of data.	

Activity 4: Structure Testing (20 minutes) – 6 marks

Test the structure and the validation of your relational database using suitable test data (normal, erroneous and extreme as appropriate).

You must provide evidence of table level testing that proves:
1. a record will not save without the customer's surname being present
2. a record will not save if the customer telephone number is not in the correct format
3. a record will not save if the customer is assigned an invalid customer type
4. a record will not save if the cost of a ticket is not valid for the type of ticket
5. a record will not save if a ticket sale does not have a valid customer
6. a record will not save if a ticket sale does not have a valid ticket type.

Complete the test log to show how you have tested the structure and validation of your database using the given **activity4.rtf** template.

Save your test log as a PDF in your folder for submission as
activity4_[Registration number #]_[surname]_[first letter of first name]

Activity 4: Structure Testing (20 minutes) – 6 marks - markscheme

Assessment focus	Band 0	Band 1	Band 2	Band 3	Max Marks
Activity 4: Structure Testing	0	1-2	3-4	5-6	6
	No rewardable material	Testing is too narrow to confirm a working solution, including limited normal, erroneous and/or extreme data. Expected results are generic or mostly inaccurate. Test data may not be present Test results prove that that the database operates under some normal circumstances relevant to the scenario. Test result comments are present when errors have been found. These comments show a limited understanding of any errors that were found.	Testing is adequate to confirm a working solution, including some normal, erroneous and/or extreme data. Expected results are mostly accurate and based on identified test data but may lack detail. Test results prove that that the database operates under some normal circumstances and that the interface can cope with some erroneous and extreme data relevant to the scenario. Test result comments are present when errors have been found. These comments show partial understanding of any errors that were found.	Testing is thorough, including a range of normal, erroneous and extreme data. Expected results are specific and accurate based on identified test data. Test results prove that that the database operates under all circumstances relevant to the scenario. Test result comments are present when errors have been found. These comments show a clear understanding of any errors and how they were fixed.	

Activity 5: Structure Evaluation (20 minutes) – 6 marks

Evaluate your database structure and validation. You should consider:

- how well your database structure has minimised data duplication
- how well your database structure meets these requirements:
- there are different types of customer. For example, a customer can be a guest of the organiser
- There are three different types of ticket:
- a Friday ticket will cost £39.00
- a Saturday ticket will cost £49.00
- a two day camping ticket will cost £88.00

Activity 5: Structure Evaluation (20 minutes) – 6 marks - markscheme

Assessment focus	Band 0	Band 1	Band 2	Band 3	Max. mark
Activity 5: Structure Evaluation	0	1-2	3-4	5-6	6
	No rewardable material	Superficial understanding of relevant technical concepts shown with some inaccuracies. Limited or unsupported justification of the relational database structure selected. Limited links between aspects of the solution and the requirements of the scenario. Technical vocabulary is used but it is not used appropriately to support arguments.	Some accurate and relevant understanding of technical concepts shown. Some valid justification, which may lack support of the relational database structure selected. Some logical links between aspects of the solution and the requirements of the scenario but may lack clarity. Mostly accurate technical vocabulary is used to support arguments.	Accurate and detailed understanding of relevant technical concepts shown throughout. A valid and fully supported justification of the relational database structure selected. Makes logical coherent links between aspects of the solution and the requirements of the scenario throughout. Fluent and accurate technical vocabulary is used to support arguments.	